

METHOD FOR CONTROLLING FILENAME DISPLAY FOR IMAGE AND VIDEO FILE TYPES

FIELD OF THE INVENTION

- 5 The present invention generally relates to graphical user interfaces (GUIs) such as those used with general-purpose computers and other types of computing devices. More specifically, the present invention generally relates to GUIs that display windows and/or folders containing computer-generated icons.

BACKGROUND OF THE INVENTION

- 10 In the computer industry, use of GUIs is well known for enabling a user to select a particular file of data (such as a word-processing file or a graphics file), from a large available selection. A GUI is a type of display format that enables a user to operate a computer by pointing to pictorial representations, such as windows and icons, displayed on a screen device. A window usually has the shape of a rectangle displayed on a screen.
- 15 Regardless of shape, a window affords a user workspace within a program and/or operating system. Generally, a user may move the window displayed on the screen, change its size and shape, enlarge it to fill the entire screen, close it entirely, or change how much of the contents of a window are displayed.

- To navigate within a GUI, such as to select a particular file to be opened, most
- 20 operating systems employ a screen cursor or pointer, typically displayed as a small arrow, that allows a user to select individual points on the screen. In operation, the cursor may be moved to a desired location in response to movements of a pointing device (e.g., a mouse,

trackball, or equivalent) by the user. In addition to affecting cursor movement, most pointing devices include one or more switches or buttons for specifying additional user input or user events. Because many user choices may be implemented through the use of a pointing device, the need for users to memorize special commands or keystrokes is lessened.

5 A standard GUI may provide access to a hierarchy of containers into which individual files can be organized. For example, a set of files may be placed in a folder, and a set of folders may be placed in another folder. This layered effect, which is associated with most general GUIs, provides for exceptional organizational features that may be utilized by a user.

10 When a folder is opened in the context of a GUI, there is typically displayed a rectangular space on the screen, and within the rectangular space is displayed a set of icons, each icon being associated with one file (such as a text or graphics file) in the folder. As previously mentioned, the rectangular space may be generally considered as a window in the context of GUIs. Once a window is opened, the contents of the window are typically
15 displayed as icons that have specific relation to folders and/or file types. For example, word-processing documents are generally all assigned icons of identical basic appearance, such as a sheet of graphically displayed paper with a folded corner, or a stylized capital W embodied within the icon. The use of similar appearing icons for similar items allows users to quickly identify the type of file that is associated with an icon. File names are also displayed adjacent
20 to icons in order to aid a user in identifying the type of file associated with the icon.

FIGURE 1 illustrates a conventional GUI window. More specifically FIGURE 1 illustrates a window 10 that contains a plurality of icons 20 in a partitioned window section 30. The window 10 also includes a supplemental window section 40 that includes menus 50 and 60. Furthermore, the window 10 also includes a standard bar area 70 and a title
25 bar 80.

As is illustrated, the window section 30 includes various icons 20 of differing appearances, each with an associated filename. For example, the icons with the associated filenames "track01.MP3" and "track02.MP3" are related to audio files, the icon associated with "outline.doc" is related to a word-processing file, and the icon with the associated
30 filename "timwpic.jpg" is related to an image file, as is the icon with the associated filename

"P01022003.jpg." The general descriptive nature of the indicated file names, along with the icons, give a user of an operating system including such a conventional window 10 the ability to determine the general contents of a file before actually opening the file.

As is illustrated in FIGURE 1, one or more files that include machine-generated file names may also be displayed. An example of a machine language generated file name is "P01022003.jpg." A machine language generated file name generally does not convey any information to the user viewing the conventional window 10. Therefore, the need to include such a file name in window 10 is open to discussion. Moreover, it is questionable whether icons in general require the use of filenames.

10 SUMMARY OF THE INVENTION

An exemplary embodiment of the present invention provides a method that renders a graphical user interface including at least one viewable graphical element, the graphical element having a filename associated therewith, the filename being absent from the user interface.

15 Another exemplary embodiment of the present invention provides a system that includes an arrangement for generating a graphical user interface including at least one viewable graphical element, the graphical element having a filename associated therewith, the file name being absent from the user interface.

Yet another exemplary embodiment of the present invention provides an article of
20 manufacture for use in programming a processor, the article of manufacture comprising at least one computer readable storage device including at least one computer program embedded therein that causes the processor to perform the method of rendering a graphical user interface including at least one viewable graphical element, the graphical element having a filename associated therewith, the filename being absent from the user interface.

25 Another exemplary embodiment of the present invention provides a method for reviewing one of a data associated with a filename and a format of the filename, the filename having an associated icon; determining if a filename is displayed based on the reviewing step; and congregating icons without displayable filenames in at least one row, the number of

icons congregated in the at least one row being greater than a comparable row with icons having filenames being displayed.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing aspects and many of the attendant advantages of this invention will
5 become more readily appreciated as the same become better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings, wherein:

FIGURE 1 illustrates a conventional window view displayable in a GUI;

FIGURE 2 illustrates a block diagram of a system suitable for providing an
10 exemplary operating environment for an exemplary embodiment of the present invention;

FIGURE 3 is a flowchart illustrating a process for illustrating icons without the presence of filenames;

FIGURE 4 illustrates a window including various icons without filenames, in accordance with an exemplary embodiment of the present invention;

15 FIGURE 5 is a flowchart illustrating a process for rendering certain icons without filenames, in accordance with another exemplary embodiment of the present invention;

FIGURE 6 illustrates a window displayed in accordance with another exemplary embodiment of the present invention;

FIGURE 7 is a flowchart illustrating a process for rendering certain icons without
20 filenames, in accordance with yet another exemplary embodiment of the present invention; and

FIGURE 8 illustrates a window displayed in accordance with yet another exemplary embodiment of the present invention.

DETAILED DESCRIPTION OF THE EXEMPLARY EMBODIMENTS

25 In order to ensure clarity in the description related to exemplary embodiments according to the present invention, the following terms are defined:

A "thumbnail" is a quantity of data that is derived from a larger quantity of data, such as a file. A thumbnail is obtained by opening one image page from a file and optically

reducing or scaling the image to a smaller size. Alternatively, discrete portions of data from a file can be pulled from the file, such as keywords, a pre-written or artificially derived summary of the data, a title, a list of column headings, any text strings that are formatted to be of larger than usual type size, which would be consistent with headlines, etc. This data
5 may be integrated with an icon to provide a user visual representation of the contents of a folder or file without having to actually open the folder or file.

A "folder" is a term which can be applied to any organizational structure that can hold one file, a plurality of files, and/or a folder or multiple folders.

An "icon" is a basic bitmap of a predetermined design, which a user can identify as
10 relating to a file, or at least a file of a certain type. In accordance with the description of the exemplary embodiments according to the present invention, an icon may or may not include associated thumbnail data.

FIGURE 2 illustrates a block diagram of a system 100 suitable for providing an exemplary operating environment for an exemplary embodiment of the present invention.
15 The system 100 illustrated in FIGURE 2 for operating an exemplary embodiment of the present invention includes a central processing unit (CPU) 110, a read-only memory (ROM) 120, and a random access memory (RAM) 130. In addition, the system 100 includes an interface 140, a display adapter 150, and an I/O adapter 160. The foregoing elements are illustrated as being generally connected and associated as one cohesive unit. However, this
20 is done by way of illustration only. The illustrated elements may be fashioned in a variety of manners well known to those skilled in the art.

The system 100 may also include various external devices. Illustrated external devices include a keyboard 170, a mouse 180, a display 190, and an external peripheral 195. The external peripheral 195 may be, for example, a printer, a storage device, etc.

25 This description of the exemplary embodiments according to the present invention will often refer to the system 100 as a computer system. However, it should be understood that the system 100 may be implemented in various operational devices. For example, the system 100, or a similar system, may be implemented in a personal digital assistant (PDA), a mobile telephone, a handheld computer, an information kiosk, etc. Therefore, it should be
30 clear that the system 100 illustrated in FIGURE 2 should be construed as exemplary and not

limiting of embodiments of the present invention. Thus, the system 100 should be as illustrative of the environment in which exemplary embodiments described herein may be used.

FIGURE 3 is a flowchart illustrating an exemplary process formed in accordance with this invention for illustrating icons without the presence of filenames. The exemplary process illustrated in FIGURE 3 includes a start block S200 that generally represents the beginning of the process illustrated in the figure. In particular, the process illustrated in FIGURE 3 may commence upon the opening of a window associated with a GUI. Opening a window within a GUI normally requires a user to interface with the system 100. In particular, a user may make a keystroke on the keyboard 170 or click on a button associated with the mouse 180, which causes the rendering of the window. Activity with, and displaying of, a window is provided as a user-interfaceable representation by way of the display 190.

Before a window is rendered on the display 190, the process illustrated in FIGURE 3 analyzes any user preferences that may be defined within the system 100 (block S210). These user preferences generally define the manner in which the window being rendered on the display 190 should display icons. That is, the preferences are examined to determine if certain icons should or should not include the use of filenames (block S220). If the preferences do not indicate that specific filenames should be removed, then a window is rendered on the display 190 that includes icons that each include the use of filenames (block S240). Such a window that includes the use of icons each having filenames is illustrated in FIGURE 1. On the other hand, if the preferences indicate that some or all of the icons should not include the use of filenames, the system 100 may render a window 400, such as that illustrated in FIGURE 4 (block S250).

FIGURE 4 illustrates a window 400 that includes various icons without filenames. The window 400 includes various icons 410 some with and some without filenames displayed in a window space 415. In particular, in the case of FIGURE 4, icons 420 and 430, depicted as multimedia (audio and/or image), icons have not been rendered with filenames. In contrast, icons 440, depicted as relating to or associated with text based files, have been rendered with associated filenames. Therefore, as is illustrated in FIGURE 4, the two audio

file icons 420 and the two image file icons 430 are illustrated without their associated filenames. However, the text-based associated icons 440 have been rendered with their associated filenames intact. Therefore, in the case of the window 400 illustrated in FIGURE 4, the preferences examined in block S210, illustrated in FIGURE 3, direct the rendering of window 400 without the use of filenames in conjunction with icons having associated with multimedia files.

Although FIGURE 3 identifies that user preferences are examined when determining the manner in which a window is rendered on the display 190, general system parameters may also dictate how icons in a window are displayed. For example, a folder or other container may be designated as a storage area for a specific type of file. In particular, it may be specifically designated by the system 100 as a folder that is to contain primarily multimedia file types. In this case, by virtue of the fact that a folder is designated as a multimedia folder type, the system may automatically dictate that icons associated with multimedia files and stored in such a designated folder will not include the use of filenames.

The preferences, whether user or system related, may be stored in various elements within the system 100. For example, user preferences may be stored and retrieved from the RAM 130 and/or the external peripheral 195 (in the event the peripheral 195 is a storage device), while system parameters may be stored in the ROM 120. The storage and retrieval of preferences related to the displaying of icon filenames pertains to each of the exemplary embodiments discussed herein.

Referring again to FIGURE 4, as is illustrated in the figure, the removal of the filenames from icons 420 and 430 creates additional space in the window space 415. In order to capitalize on this additional space obtained with the removal of filenames from icons 420 and 430, the computer system 100 may position the icons 410 in closer proximity to each other. This may result in an increased number of icons being displayable in the window space 415.

In the case of the icons 410 displayed in the window space 415, both horizontal and vertical adjustments may be made to increase the amount of space available to display icons 410 in the window space 415. For example, icons 420 and 430, in the top row, may be shifted closer together, since the absence of filenames prevents the possibility that filenames

from adjacent icons will overlap. Furthermore, the second row of icons 410 may be shifted upward, as a group, in order to facilitate rendering an additional row of icons in the same window space 415. Clearly, based upon the number of filenames removed, horizontal and/or vertical shifting of icons may be made in many different ways.

5 Another repositioning technique may also be implemented in conjunction with the exemplary embodiments of the present invention. In particular, in the case of the icons displayed in FIGURE 4, it may be desirable to position the icon 430 in the second row adjacent to an icon that has also had its filename removed. For example, the mentioned icon 430 may be repositioned adjacent to any one of the other icons that had filenames removed.

10 Both this repositioning technique along with the technique described in the preceding paragraph may be used alone or in combination together. Moreover, the described techniques may be used with each of the exemplary embodiments described herein.

FIGURE 5 shows a flowchart illustrating a process for rendering certain icons without filenames, in accordance with another exemplary embodiment of the present invention. As is illustrated in FIGURE 5, the exemplary embodiment includes a start block S500 that generally represents the beginning of the process illustrated in the figure. In particular, the process illustrated in FIGURE 5 may commence upon the opening of a folder associated with a GUI. Opening a window within the GUI normally requires a user to interface with the system 100. In particular, a user may make a keystroke on the

15 keyboard 170 or click on a button associated with the mouse 180, which causes the rendering of the window. Activity with, and displaying of, a window is provided as a user-interfaceable representation by way of the display 190.

Before a window is rendered on the display 190, the process illustrated in FIGURE 5 analyzes any user preferences that may be defined within the system 100 (block S510).

25 These user preferences generally define the manner in which the window being rendered on the display 190 should display icons. That is, the preferences are examined to determine if certain icons should not include the use of machine generated filenames (block S520). If the preferences do not indicate that specific filenames should be removed, then a window is rendered on the display 190 that includes icons that each include the use of filenames (block S530), or renders icons without filenames based upon other criteria. Such a window

30

that includes the use of icons each having filenames is illustrated in FIGURE 1. On the other hand, if the preferences indicate that the icons should not include the use of machine generated filenames, the system 100 may render a window 600, such as that illustrated in FIGURE 6 (block S540).

5 FIGURE 6 illustrates a window displayed in accordance with another exemplary embodiment of the present invention. FIGURE 6 illustrates the window 600 that includes various icons 610 with or without filenames displayed in a window space 615. In particular, in the case of FIGURE 6, an icon 620 has not been rendered with a filename. However, as is illustrated in FIGURE 6, all other icons have been rendered with associated filenames
10 maintained. The icon 620 was rendered without its associated filename because the filename is machine generated. The machine generated icon can be seen in FIGURE 1. Therefore, in the case of the window 600 illustrated in FIGURE 6, the preferences examined in block S510, illustrated in FIGURE 5, direct the rendering of window 600 without the use of filenames in conjunction with icons having associated machine generated filenames.

15 FIGURE 7 a flowchart illustrating a process for rendering certain icons without filenames, in accordance with yet another exemplary embodiment of the present invention. As is illustrated in FIGURE 7, the exemplary embodiment includes a start block S700 that generally represents the beginning of the process illustrated in the figure. In particular, the process illustrated in FIGURE 7 may commence upon the opening of a folder associated with
20 a GUI. Opening a window within the GUI normally requires a user to interface with the system 100. In particular, a user may make a keystroke on the keyboard 170 or click on a button associated with the mouse 180, which causes the rendering of the window. Activity with, and displaying of, a window is provided as a user-interfaceable representation by way of the display 190.

25 Before a window is rendered on the display 190, the process illustrated in FIGURE 7 analyzes any user preferences that may be defined within the system 100 (block S710). These user preferences generally define the manner in which the window being rendered on the display 190 should display icons. That is, the preferences are examined to determine if certain icons should not include the use of filenames (block S720). If the preferences do not
30 indicate that specific filename types should be removed, then a window is rendered on the

display 190 that includes icons that each include the use of filenames (block S740), or renders icons without filenames based upon other criteria. Such a window that includes the use of icons each having filenames is illustrated in FIGURE 1. On the other hand, if the preferences indicate that the certain icons should not include the use of filenames, the system 100 may render a window 700, such as that illustrated in FIGURE 8 (block S750). In this case, the preferences indicate JPEG filenames, or image files, should not be displayed with associated icons (see FIGURE 8).

FIGURE 8 illustrates a window displayed in accordance with yet another exemplary embodiment of the present invention. FIGURE 8 illustrates a window 800 that includes various icons 810 with or without filenames displayed in a window space 815. In particular, in the case of FIGURE 8, icons 820 have not been rendered with a filenames. However, as is illustrated in FIGURE 8, all other icons have been rendered with associated filenames maintained. The icons 820 were rendered without the associated filename, respectively, because the each filename relates to an image (based upon .jpg extension). The JPEG icons with filenames can be seen in FIGURE 1. Therefore, in the case of the window 800 illustrated in FIGURE 8, the preferences examined in block S710, illustrated in FIGURE 7, direct the rendering of window 800 without the use of filenames in conjunction with icons having associated image filenames.

While the exemplary embodiments of the invention have been illustrated and described, it will be appreciated that various changes can be made therein without departing from the spirit and scope of the present invention.